Report on Computer Security Track for ONR Workshop on High Assurance Computing

Catherine Meadows
Center for High Assurance Computing Systems
Naval Research Laboratory
Washington, DC 20375
USA

November 13, 1995

1 Introduction

This report gives an account of the discussion of the security track that was held as part of the ONR Workshop on High Assurance Computing at the Naval Research Laboratory February 1995. The security track was chaired by Teresa Lunt, ARPA and Catherine Meadows, NR:. Members of the track were Emilie J. Siarkiewicz, Rome Labs, John McLean, NRL, Virgil Gligor, UofMd, Dick Kemmerer, UCSB, John McHugh, Portland State Univ., Ravi Sandhu, George Mason Univ., Sue Rho, TIS, Helen Gill, NSF, Mary Bernstein, TIS, Michael Reiter, AT&T Bell Labs, John Van Tassell, NSA, and Randy Johnson, NSA.

2 Overview of Computer Security

Computer security may be said to be the guarantee of critical system properties in face of possible attack by an intelligent, hostile, intruder. The property most generally associated with security is confidentiality, but integrity and guarantee of service are also considered security properties.

Research in computer security covers a wide range of areas. Probably the oldest of these is cryptography: the design of methods to guarantee secrecy of data in transit though an insecure environment. Areas of research in cryptography include not only the design and analysis of encryption algorithms, including public and private key algorithms, signature schemes, secure hash functions, and cryptographically secure random number generators, but the design and analysis of protocols that use cryptography for applications such as cryptographic key distribution, authentication, and secure electronic transactions.

Cryptography, of course, predates computer security by several thousand years, and until fairly recently, computer security and cryptography moved in parallel but nonintersecting tracks. Cryptographic research concentrated on security of communications, while computer security research concentrated on securing standalone operating systems. However, with the growing popularity and dependence upon distributed systems and networks, this has changed. Now most secure systems have requirements both for the use of secure cryptoalgorithms and protocols, and for the securing of individual components of the system. Often, the term "information security" is used as an umbrella term to cover both cryptography and computer security, but for the purposes of this report we will assume that computer security includes cryptography.

A great deal of work has also been done on access controls in computer systems. This work is generally divided into access controls that can enforce a wide variety of policies, versus those that enforce what is known as mandatory access control, which is a policy associated with multilevel secure systems. MLS sytems are government systems that must store data classified at different security levels and allow users to access only what they are cleared for.

Work in multilevel security has also encompassed work in the areas of information flow and covert channel analysis and prevention. In most multilevel architectures, a small portion of the system is trusted to support the security policy. The rest is untrusted, and it is assumed that it may actually contain malicious "Trojan Horse" code that may attempt to subvert the security policy of the system. In particular, if an untrusted Trojan Horse process with access to information classified at a high level is able to affect the system in a way visible to an untrusted Trojan Horse process without such access, this fact may be used to set up a communication channel by which the high process may communicate classified information to the low process, thus subverting the access controls. Such illicit channels are called *covert channels*, and have been the subject of much study in research in multilevel secure systems. Techniques have been developed for discovering, measuring, and controlling these channels, and a good deal of work has also gone into developing *information flow models* that can be used to design systems that are free of certain classes of channels.

Another area in which research has been active is that of intrusion detection. This is the technique of using analysis of audit data to determine whether or not an intrusion into a computer system (by either a human or a virus) has occurred. Challenges here include dealing with the vast amount of audit data, processing the data in a reasonable amount of time, and correctly identifying intruders while avoiding the incorrect identification of honest users as intruders.

An area that has emerged recently is that of the formal modeling and analysis of policies. As computers become used in more and more applications, the number and types of access control policies that must be enforced has grown, and these policies are often complex. When the system is implemented, it is often an imperfect understanding of the policy or gaps and inconsistencies in the policy itself that cause security failures. The formal modeling and analysis of security policies help system designers understand the policy better and identify possible problems.

A considerable amount of work has also gone into developing engineering techniques for secure systems as well. These include the exploration of different types of architectures, including specialized hardware and microkernels, the conversion of existing commercial operating systems to secure ones, and the development of engineering techniques for assuring that the system is trustworthy. In this last, formal methods has traditionally played an important role in guaranteeing the trustworthiness in multilevel systems that require a high degree of assurance. Indeed, much of the research in formal methods has been funded by the security community.

Much of the earlier work in computer security concentrated on operating systems. Although this is still an active area, the focus is now shifting to such areas as application security (in particular security of databases), secure systems which must satisfy other critical properties besides security (for example, safety or real-time), and integration of secure systems into larger environments. These not only raise issues concerning the security each component, but concerning the ways in which different compenents rely upon each other in order to enforce the security policy, and the ways in which the components enforcing the security policy interact with the rest of the system.

Although research has been active in all of the areas described above, not as much of it is being used in actual systems as might be expected. The Internet, for example, has a number of security holes that have been known for years, and many could be fixed with existing security techniques. There are several reasons for this state of affairs. One is that security adds cost to a system, both in time and effort in development, evaluation, and system performance. Security researchers have been seen traditionally as treating security as paramount and not paying enough attention to the effects that introducing security has on the other system requirements. Thus security may be left out of a system as a cost-cutting measure. It is also the case that a system may be implemented with security controls appropriate for its initial intended use, but either new ways of using the system or unforseen types of attacks make the security controls inadequate. Once a system has been fielded without appropriate security controls, it is very difficult to retrofit it, and this is what causes many of the problems of introducing security into systems today. The Internet is one such example. It was originally intended for communication among a relatively small number of scientists, but it now handles a large amount of traffic and is being considered for many more applications such as electronic banking. Another example is the virus problem in personal computers. PCs were originally developed without even rudimentary security controls, since they were intended for individual use, a fact that now makes it possible for computer viruses to flourish.

One counterexample to this rather pessimistic picture is the firewall, which has been used to add at least a measure of security to a number of existing systems. Briefly, a firewall is component of a system that polices all traffic going into and out of the system, and only allows communications that are considered "safe." Firewalls have an advantage in that they are relatively straightforward to install and require minimal modifications of existing systems. Morever, they can be installed unilaterally without waiting for the support of a more general security infrastructure. They also have disadvantages in that they reduce the ease with which users of the system may communicate with the outside world, since many useful types of communication are considered unsafe from the security point of view, and the security they provide is limited since they can only prevent attacks aimed at certain levels of the protocol hierarchy [1]. However, although firewalls have a number of disadvantages, their popularity can give us some insight into what kinds of security solutions will be accepted by the community at large.

3 Example Systems

In order to focus the discussion at the workshop, we identified three example systems with different kinds of security needs. These systems' security requirements illustrate both the areas of research and the security issues that we identified in the previous section. In order to bring out the point that security must take into account other system requirements, for each system we identified a "showstopper", that is, a requirement that, if interfered with by a security solution, would probably cause that solution to be rejected.

These systems are listed below.

3.1 Medical Information Systems

As medical records become more and more computerized, a number of new security issues arise. A computerized medical information system must protect the patients' privacy, while allowing those who have legitimate access to portions of the data (such as physicians, nurses, insurance agents) to view portions of it. It must guard against unauthorized and possibly malicious alterations of patients' records. Any such policy it does enforce must be flexible, since changes in the law can affect who is authorized to access what data. It must also be possible to override the security controls in the case of emergency; in these cases it is important that an audit trail be provided. However, any extra documentation in these instances should not be too onerous.

Some of the challenges are: the vast amount of information that needs to be processed, the long life of the patient records, which may mean that records will be stored on different types of media during a patient's lifetime, the informal manner in which patient records are organized, the need for different kinds of information by different organizations and indivduals, and the ambiguity and vagueness of current laws and policies.

SHOWSTOPPER: Anything that interferes with the timely delivery of information in an emergency.

4 The Cellular Telephone Infrastructure

Cellular telephone service is currently provided in the following way. Each telephone has a unique Electronic Serial Number, or ESN. When a phone requests a service, it sends its ESN to the service provider, who verifies that this is a valid number, and provides the service. If the provider is the phone's home provider, it bills the owner of the phone for the service. If the provider is not the phone's home provider, it bills the home provider, who bills the owner of the phone. If for some reason the bill is in error (that is, the service was obtained fraudulently) then the home provider is still responsible for the bill.

The problem here is that all authenticating information is sent over air channels in the clear. Thus it is a trivial matter for a thief to copy an ESN off the air and program it into his own phone. This "cloned" phone can then be used to impersonate a legitimate subscriber. Since, if a phone is "roaming", the home provider is still responsible for all bills sent by the visited provider, this has caused serious losses to the cellular phone industry.

A number of solutions have been proposed, and some are already in use. These include the use of intrusion detection software, the use of methods developed by the military to identify radio "fingerprints" of phones, and the use of PINS that are entered by the user and transmitted over a different channel than the ESN (to make it harder for a thief to match up PIN with ESN). A set of standards for authentication methods and protocols is also under development.

The problem here is to produce some means of detecting or preventing fraud that is cost-effective to implement and that will not interfere with providing service to a legitimate customer. It should take into account the fact that different providers will have different resources and needs and may not all want to implement controls at the same degree of strength. Thus, providers should be given options as to the degree of security they provide, and a provider's decision to minimize its own protection should not threaten the protection given to other providers.

SHOWSTOPPER: Anything that interferes with a provider providing service to a legitimate customer.

5 JMCIS (JOINT MARITIME COMMAND INFORMATION SYSTEM)

This is a system providing a common operating environment for Navy Command and Control Systems. Architecturally, it consists of a message server talking to a communication network and the JMCIS LAN, which talks to a Command Database and the JMCIS Client. The current accredited interconnection between two systems at different levels (such as SECRET and TOP SECRET) involves

1. A sanitizer sitting between message servers for communication from high to low. This is monitored by a human.
2. A line from the low message server to the high message server that allows the low server to send information to the high one, but does not allow the high server to acknowledge receipt, for fear of covert channels.

This setup is not optimal for several reasons:

1. The sanitization process is slow and risky (it is not clear that the human can catch everything, particularly if we assume the possibility of Trojan Horses in the high system).
2. Since acknowledgements are not allowed, the channel from low to high is not that reliable.
3. Message servers cannot handle anything other than Naval messages. If different kinds of data are generated within the low system, and it needs to be sent to the high one, it must by done by putting the data on a floppy disk and handcarrying it.

The problems are to:

1. Reduce the risk of downgrading;
2. Increase the efficiency of downgrading;
3. Increase the efficiency and reliability of sending information from low to high without, and; compromising security
4. Do all of this without requiring a lengthy reaccreditation process

An account of some different solutions to the problem is given in [2]/

SHOWSTOPPER: Anything that makes the accreditation process difficult.

6 Report on Discussion: General Security Issues

In the discussions with the three other tracks, a number of issues in security research came up again and again. We list them separately here.

It seemed that one of the most important contributions that security had to make was the understanding that it brought to the results of a hostile attack underlying other system properties. In the other areas, assumptions about the system environment were either that is was benign or dangerous, but not malicious. System properties that hold under these conditions might not hold up under hostile attack. Thus an understanding of security gives the opportunity to put these assumptions through a "stress test."

The issue of assurance was very important, in security as well as other areas. In particular, it is necessary to be able to identify the critical and non-critical parts of the system, have methods of assurance based on compositionality, so that we know what properties are possessed by a system whose components satisfy certain properties, and to be able to develop convincing assurance arguments that support certification and accreditation of secure systems.

Security research has traditionally concentrated on confidentiality of data. But we need the ability to model and reason about a wide range of security properties. This became especially clear as we saw the many different ways in which security could contribute to other fields.

Security often prove to be in conflict with other critical system properties. Thus a framework for reasoning about tradeoffs with other system properties was needed. An understanding of how a system developed to satisfy other requirements besides security could be reengineered to include security was also needed.

Finally, although this was not exactly a research issue, the issue of technology transfer cut through all four tracks. Better technology transfer is needed, not only so that our technology will see use, but so that researchers can get better and earlier feedback on what works and what doesn't.

7 Report on Discussion: Interactions with Other Tracks

7.1 Real-time Track

Of the three types of requirements: real-time, safety, and fault-tolerance, it is real-time that comes most often into conflict with security. Any kind of security mechanism will tend to slow down a system, thus interfering with real-time requirements. What often happens in the implementation of a system with both security and real-time requirements is that security mechanisms are eliminated when they are found to interfere with the more pressing real-time requirements. Thus one of the areas in which it was agreed that more research was needed was the development of some kind of framework for tradeoff analysis so that the conflicts between security and real-time could be identified up front and appropriate choices could be made early in the system design.

One area in which it was unclear whether or not a tradeoff framework would be helpful is in the area of covert channel analysis. As Nancy Lynch pointed out in her presentation, the goal of real-time is not so much to make system behavior fast, but to make it predictable. This may include requirements that a system response occur within a certain window of time. Many approaches to reducing the capacity of covert channels, however, do so by making the ways in which a high process can affect the system more unpredictable. This goes beyond the usual sort of tradeoffs that are considered in the building of real-time systems. The problem of reasoning about tradeoffs between predictability and unpredictability would thus require additional research.

In any case, the ability to reason about tradeoffs requires the ability to develop some sort of metric. This is fairly straightforward in the case of real-time requirements, but has proved to be more difficult for security. Thus the development of some meaningful set of metrics for security is another possible area of investigation.

Although security is often in conflict with real-time requirements, it was noted that in several cases security methods themselves rely on time in some way. An example cited was the use of timestamps in cryptographic protocols. There was not sufficient time to go into this in more detail, but it would probably be worthwhile to consider the various ways in which time is used in security, and how research in real-time systems could be brought to bear on this.

It was also noted that many of the assumptions behind research in real-time systems do not account for the actions of a hostile intruder who is trying to make the system fail. Another area of research could be to see how to make a system satisfy real-time requirements in the face of a hostile intruder who is trying to prevent this from happening. This would be a stronger requirement than the usual denial-of-service issues that security research now focuses on.

Finally, it was noted that the formal methods used in security and real-time research appear to have much in common. Both security and real-time have used state exploration techniques to prove desirable properties. It would be worth while to see if there are any important differences in what has been developed, and if techniques that have been developed for one application could be leveraged by the other.

7.2 Safety

Of the three areas real-time, safety, and fault-tolerance, security has the most in common with safety. Both are concerned with preventing bad things from happening. The main difference appears to be the assumptions that are made about the environment. Safety research generally assumes that the environment may be dangerous, but not hostile or intelligent. Security generally assumes that the environment includes a hostile, intelligent, adversary. This distinction may become blurred, however. For example, in the design of software controlling a nuclear power plant, it may be necessary to take into account the possible existence of a terrorist who is trying to cause the plant to destroy itself and its environment. Since the system is already being designed to prevent failures caused by natural means, the best design strategy is to extend the safety model to include the hostile intruder, rather than to introduce this as a separate security concern. Thus safety and security may be seen as part of a continuum.

In spite of the fact that safety and security have so much in common, they have operated in very different traditions. In general, security research has tended to focus mainly on a few well-defined areas and threats, gaining more depth than breadth, while safety research focuses on making a system safe from all kinds of hazards, thus gaining more breadth than depth. Thus the two areas probably have much to learn from studying each other's methods.

In particular, the safety researchers at the meeting were interested in learning about the security researchers' experience (both good and bad) with the security kernel, a means by which one small centralized part of the system is given the responsibility of enforcing all accesses to data. Is it possible that a "safety kernel" could be built along the same lines? Under what cases would it be feasible to build a safety kernel, and under what cases not?

Conversely, an area of safety research that aroused the interest of the security researchers was that of hazard analysis, a means for identifying (and thus designing against) the hazards that a system may be subject to. Too often, a secure system is designed to operated only under a vary narrow range of conditions, and when these conditions are violated, it is hard to understand what kinds of protection the system provides. Thus, in many cases, system break-ins result, not from compromise of the security mechanisms themselves, but from taking advantage of the unsafe ways in which the system is used. A hazard analysis done upfront could avoid these situations or at least give the user of the system a warning that the system is not being used in a safe way.

7.3 Fault-Tolerance

In its relationship to security, fault-tolerance appears to fall between real-time and safety. Instead of being most often directly in conflict with security, like real-time, or another point on the continuum, like safety, fault-tolerance seems to be sometimes in conflict with security, but more often complementary. In many cases the fault-tolerant properties of a system will rest upon the assumption the system will have certain security properties, while the security properties of a system will rest upon the assumption that it will have certain fault-tolerant properties. A concern was raised that in some cases these dependencies might be circular, and that one of the problems facing research in these areas would be to identify and eliminate these dependencies. In particular, a better understanding of how security techniques could be used to support the integrity of information needed to guarantee fault-tolerance was needed. Many algorithms for fault-tolerance assume that the information is protected against hostile modification, but not much thought has gone into how this protection is provided.

As for conflicts between fault-tolerance and security, it was noted that most security requirements focus on providing confidentiality and integrity, while most fault-tolerance requirements focus on providing availability. These can often be in conflict, since one of the most straightforward ways of guaranteeing the first two properties is to restrict access to the system when they are threatened. Thus, in many cases, a system may be designed to be highly fault-tolerant, but security requirements will be ignored, or it may be designed to be highly secure, but availability will be restricted. An example is the cellular phone infrastructure described above, which was designed to be highly available, but was originally implemented with almost no security controls. Thus the issue of retrofitting security into fault-tolerant systems and vice versa was raised. In order to do this, we need to understand both when security and fault-tolerance come into conflict, and when they can be made to support one another.

Finally, the question of a challenge problem to integrate fault-tolerance and security was discussed. One possibility that was mentioned was a highly available, authentication subnet of the Internet, which could be use, for example, for electronic transactions.

References

[1] William Cheswick and Steven Bellovin. *Firewalls and Internet Security: Repelling the Wily Hacker.* Addison-Wesley, 1994.

[2] J. N. Froscher, D. M. Goldschlag, M. H. Kang, C. E. Landwehr, A. P. Moore, I. S. Moskowitz, and C. N. Payne. Improving inter-enclave information flow for a secure strike planning operation. In *Proceedings of the 11th Annual Computer Security Applications Conference*, December 1995.